A Partition-Based Relaxation For Steiner Trees

Jochen Könemann* David Pritchard* Kunlun Tan*

February 2, 2008

Abstract

The Steiner tree problem is a classical NP-hard optimization problem with a wide range of practical applications. In an instance of this problem, we are given an undirected graph $G = (V,E)$, a set of *terminals* $R \subseteq V$, and non-negative costs c_e for all edges $e \in E$. Any tree that contains all terminals is called a *Steiner tree*; the goal is to find a minimum-cost Steiner tree. The nodes $V \setminus R$ are called *Steiner nodes*.

The best approximation algorithm known for the Steiner tree problem is due to Robins and Zelikovsky (SIAM J. Discrete Math, 2005); their *greedy* algorithm achieves a performance guarantee of $1 + \frac{\ln 3}{2} \approx 1.55$. The best known *linear programming* (LP)-based algorithm, on the other hand, is due to Goemans and Bertsimas (Math. Programming, 1993) and achieves an approximation ratio of $2 - 2/|R|$. In this paper we establish a link between greedy and LP-based approaches by showing that Robins and Zelikovsky's algorithm has a natural primal-dual interpretation with respect to a novel *partition*-based linear programming relaxation. We also exhibit surprising connections between the new formulation and existing LPs and we show that the new LP is stronger than the bidirected cut formulation.

An instance is *b-quasi-bipartite* if each connected component of $G \setminus R$ has at most b vertices. We show that Robins' and Zelikovsky's algorithm has an approximation ratio better than $1 + \frac{\ln 3}{2}$ for such instances, and we prove that the integrality gap of our LP is between $\frac{8}{7}$ and $\frac{2b+1}{b+1}$.

1 Introduction

The Steiner tree problem is a classical problem in combinatorial optimization which owes its practical importance to a host of applications in areas as diverse as VLSI design and computational biology. The problem is NP-hard [21], and Chlebík and Chlebíková show in [6] that it is NP-hard even to *approximate* the minimum-cost Steiner tree within any ratio better than $\frac{96}{95}$. They also show that it is NP-hard to obtain an approximation ratio better than $\frac{128}{127}$ in *quasi-bipartite* instances of the Steiner tree problem. These are instances in which no two Steiner vertices are adjacent in the underlying graph G.

1.1 Greedy algorithms and r-Steiner trees

One of the first approximation algorithms for the Steiner tree problem is the well-known *minimum-spanning tree heuristic* which is widely attributed to Moore [14]. Moore's algorithm has a performance ratio of 2 for the Steiner tree problem and this remained the best known until the 1990s, when Zelikovsky [41] suggested computing Steiner trees with a special structure, so called *r-Steiner trees*. Nearly all of the Steiner tree algorithms developed since then use r-Steiner trees. We now provide a formal definition.

* Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada. Email: {jochen,dagpritc,ktan}@math.uwaterloo.ca

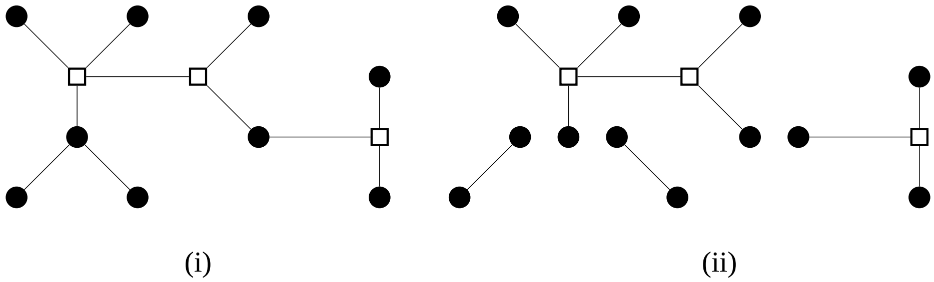

Figure 1: The figure shows a Steiner tree in (i) and its decomposition into full components in (ii). Square and round nodes correspond to Steiner and terminal vertices, respectively. This particular tree is 5-restricted.

A *full Steiner component* (or *full component* for short) is a tree whose internal vertices are Steiner vertices, and whose leaves are terminals. The edge set of any Steiner tree can be partitioned into full components, by *splitting* the tree at terminals: see Figure 1 for an example. An *r-(restricted)-Steiner tree* is defined to be a Steiner tree all of whose full components have at most r terminals. We remark that such a Steiner tree may in general not exist; for example, if G is a star with a Steiner vertex at its center and more than r terminals at its tips. To avoid this problem, each Steiner vertex v is *cloned* sufficiently many times: introduce copies of v and connect these copies to all of v's neighbors in the graph. Copies of an edge have the same cost as the corresponding original edge in G.

Let $\mathtt{opt}$ and $\mathtt{opt}_r$ be the cost of an optimum Steiner tree and that of an optimal r-Steiner tree, respectively, for the given instance. Define the *r-Steiner ratio* ρ_r as the supremum of $\mathtt{opt}_r/\mathtt{opt}$ over all instances of the Steiner tree problem. In [5], Borchers and Du provided an exact characterization of ρ_r. The authors showed that $\rho_r = 1 + \Theta(1/\log r)$ and hence that ρ_r tends to 1 as r goes to infinity.

Computing minimum-cost r-Steiner trees is NP-hard for $r \geq 4$ [13], even if the underlying graph is quasi-bipartite. The complexity status for $r = 3$ is unresolved, and the case $r = 2$ reduces to the minimum-cost spanning tree problem.

In [41], Zelikovsky used 3-restricted full components to obtain an $11/6$-approximation for the Steiner tree problem. Subsequently, a series of papers (e.g., [4, 20, 22, 30]) improved upon this result. These efforts culminated in a recent paper by Robins and Zelikovsky [34] in which the authors presented a $\left(1+\frac{\ln 3}{2}\right) \approx 1.55$-approximation (subsequently referred to as $\mathtt{RZ}$) for the r-Steiner tree problem. They hence obtain, for each fixed $r \geq 2$, a $1.55\rho_r$ approximation algorithm for the (unrestricted) Steiner tree problem. We refer the reader to two surveys in [19, 31].

1.2 Approaches based on linear programs

There is a large body of work on linear programming (LP)-based approximation algorithms for problems in combinatorial optimization. First, one finds a *good* LP relaxation for the problem. Then one designs an algorithm that produces a feasible integral solution whose cost is provably close to that of an optimum fractional solution for this relaxation. Many aspects of different LP relaxations for the Steiner tree problem have been investigated (e.g., [3, 8, 9, 10, 12, 17, 27, 38, 39]).

Many of these LPs have been fruitfully used in *integer programming*-based approaches to exactly solve instances of up to ten thousand nodes [28]. Another common area in which LPs are useful is the design of polynomial time approximation algorithms via the *primal-dual method* (e.g., [18]). In this method, a feasible solution of the relaxation's LP dual is used to obtain a lower bound on the optimum cost.

The "classical" LP-based approximation algorithms for Steiner trees [16] and forests [2] use the *undirected cut relaxation* [3] and have a performance guarantee of $2 - \frac{2}{|R|}$. This relaxation has an integrality gap of $2 - \frac{2}{|R|}$ and the analysis of these algorithms is therefore tight. Slightly improved algorithms have since

been designed [23, 26] but do not achieve any constant approximation factor better than 2.

In the special case of quasi-bipartite graphs, Rajagopalan and Vazirani [32] and Rizzi [33] obtained a $\frac{3}{2}$ approximation for the Steiner tree problem in quasi-bipartite graphs. The analysis of [32] applies the primal-dual method to the *bidirected cut relaxation* [12, 39]. The bidirected cut relaxation is widely conjectured to have a worst-case integrality gap that is close to 1: the worst known example shows a gap of only $\frac{8}{7}$ (see Section 5). Despite its conjectured strength, this new relaxation has not yet given rise to a Steiner tree algorithm with performance guarantee better than 2 in general graphs.

1.3 Contribution of this paper

In this paper we provide algorithmic evidence that the primal-dual method is useful for the Steiner tree problem. We first present a novel LP relaxation for the Steiner tree problem. It uses full components to strengthen a formulation based on *Steiner partition* inequalities [8]. We then show that the algorithm $\mathtt{RZ}$ of Robins and Zelikovsky can be analyzed as a primal-dual algorithm using this relaxation. We can show (see Section 5) that our relaxation is strictly stronger than the *standard* Steiner partition formulation; so the use of full components strengthens the partition inequalities.

In [34], Robins and Zelikovsky showed that $\mathtt{RZ}$ has a performance ratio of 1.279 for quasi-bipartite graphs, and a performance ratio of 1.55 in general graphs. We prove a natural interpolation of these two results. For a Steiner vertex v, define its *Steiner neighborhood* S_v to be the collection of vertices that are in the same connected component as v in $G\backslash R$. A graph is *b-quasi-bipartite* if all of its Steiner neighborhoods have cardinality at most b. Note, “1-quasi-bipartite” is synonymous with “quasi-bipartite.” We prove:

Theorem 1. *Given an undirected, b-quasi-bipartite graph $G=(V,E)$, terminals $R\subseteq V$, and a fixed constant $r\geq 2$, Algorithm $\mathtt{RZ}$ returns a feasible Steiner tree T s.t.*

$$c(T)\leq\begin{cases}1.279\cdot\mathtt{opt}_r & : \quad b=1\\ (1+\frac{1}{e})\cdot\mathtt{opt}_r & : \quad b\in\{2,3,4\}\\ \left(1+\frac{1}{2}\ln\left(3-\frac{2}{b}\right)\right)\mathtt{opt}_r & : \quad b\geq 5.\end{cases}$$

Unfortunately, Theorem 1 does not imply that our new relaxation has a small integrality gap. Nonetheless, we obtain the following bounds, when G is b-quasi-bipartite:

Theorem 2. *Our new relaxation has an integrality gap between $\frac{8}{7}$ and $\frac{2b+1}{b+1}$.*

2 Spanning trees and a new LP relaxation for Steiner trees

Our work is strongly motivated by, and uses, results on the spanning tree polyhedron due to Chopra [7]. In this section, we first discuss Chopra’s characterization of the spanning tree polyhedron; then we mention a primal-dual interpretation of Kruskal’s spanning tree algorithm [25] based on Chopra’s formulation. Finally we extend ideas in [8, 9] to derive a new LP relaxation for the Steiner tree problem.

2.1 The spanning tree polyhedron

To formulate the minimum-cost spanning tree (MST) problem as an LP, we associate a variable x_e with every edge $e\in E$. Each spanning tree T corresponds to its *incidence vector* x^T, which is defined by $x^T_e=1$ if T contains e and $x^T_e=0$ otherwise. Let Π denote the set of all partitions of the vertex set V, and suppose that $\pi\in\Pi$. The *rank* $r(\pi)$ of π is the number of parts of π. Let E_π denote the set of edges whose ends lie in different parts of π. Consider the following LP.

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e && (\mathrm{P}_{SP}) \\ \text{s.t.} \quad & \sum_{e \in E_\pi} x_e \geq r(\pi) - 1 \quad \forall \pi \in \Pi, \\ & x \geq 0. \end{aligned}$$

Chopra [7] showed that the feasible region of (P_{SP}) is the convex hull of all incidence vectors of spanning trees, and hence each basic optimal solution corresponds to a minimum-cost spanning tree. Its dual LP is

$$\begin{aligned} \max \quad & \sum_{\pi \in \Pi} (r(\pi) - 1) \cdot y_\pi && (\mathrm{D}_{SP}) \\ \text{s.t.} \quad & \sum_{\pi : e \in E_\pi} y_\pi \leq c_e \quad \forall e \in E, && (1) \\ & y \geq 0. && (2) \end{aligned}$$

2.2 A primal-dual interpretation of Kruskal's MST algorithm

Kruskal's algorithm can be viewed as a continuous process over *time*: we start with an empty tree at time 0 and add edges as time increases. The algorithm terminates at time τ^* with a spanning tree of the input graph G. In this section we show that Kruskal's method can be interpreted as a primal-dual algorithm (see also [18]). At any time $0 \leq \tau \leq \tau^*$ we keep a pair (x_τ, y_τ), where x_τ is a partial (possibly infeasible) 0-1 primal solution for (P_{SP}) and y_τ is a feasible dual solution for (D_{SP}). Initially, we let $x_{e,0} = 0$ for all $e \in E$ and $y_{\pi,0} = 0$ for all $\pi \in \Pi$.

Let G_τ denote the forest corresponding to partial solution x_τ and let E_τ denote its edges, i.e., $E_\tau = \{e \in E \mid x_{e,\tau} = 1\}$. We then denote by π_τ the partition induced by the connected components of G_τ. At time τ, the algorithm then increases y_{π_τ} until a constraint of type (1) for edge $e \in E \setminus E_{\pi_\tau}$ becomes tight. Assume that this happens at time $\tau' > \tau$. The dual update is

$$y_{\pi_\tau, \tau'} = \tau' - \tau.$$

We then include e in our solution, i.e., we set $x_{e,\tau'} = 1$. If more than one edge becomes tight at time τ', we can process these events in any arbitrary order. Thus, note that we can pick any such tight edge first in our solution. We terminate when G_τ is a spanning tree. Chopra [7] showed that the final primal and dual solutions have the same objective value (and are hence optimal), and we give a proof of this fact for completeness.

Theorem 3. *At time τ^*, algorithm* `MST` *finishes with a pair (x_{τ^*}, y_{τ^*}) of primal and dual feasible solutions to* (P_{SP}) *and* (D_{SP})*, respectively, such that*

$$\sum_{e \in E} c_e x_{e,\tau^*} = \sum_{\pi \in \Pi} (r(\pi) - 1) \cdot y_{\pi,\tau^*}.$$

Proof. Notice that for all edges $e \in E_{\tau^*}$ we must have $c_e = \sum_{\pi : e \in E_\pi} y_{\pi,\tau^*}$ and hence, we can express the cost of the final tree as follows:

$$c(G_{\tau^*}) = \sum_{e \in E_{\tau^*}} \sum_{\pi : e \in E_\pi} y_{\pi,\tau^*} = \sum_{\pi \in \Pi} |E_{\tau^*} \cap E_\pi| \cdot y_{\pi,\tau^*}.$$

By construction the set $E_{\tau^*} \cap E_\pi$ has cardinality exactly $r(\pi) - 1$ for all $\pi \in \Pi$ with $y_{\pi,\tau^*} > 0$. We obtain that $\sum_{e \in E} c_e x_{e,\tau^*} = \sum_{\pi \in \Pi} (r(\pi) - 1) \cdot y_{\pi,\tau^*}$ and this finishes the proof of the lemma. □

Observe that the above primal-dual algorithm is indeed Kruskal's algorithm: if the algorithm adds an edge e at time τ, then e is the minimum-cost edge connecting two connected components of G_τ.

2.3 A new LP relaxation for Steiner trees

In an instance of the Steiner tree problem, a partition π of V is defined to be a *Steiner partition* when each part of π contains at least one terminal. Chopra and Rao [8] introduced this notion and proved that, when x is the incidence vector of a Steiner tree and π is a Steiner partition, the inequality

$$\sum_{e \in E_\pi} x_e \geq r(\pi) - 1. \tag{3}$$

holds. These *Steiner partition inequalities* motivate our approach.

In the following we use $G[U]$ to denote the subgraph of G induced by vertex set U, i.e., the graph with vertex set U and such that $E(G[U]) = \{uv \in E(G) \mid u \in U, v \in U\}$. We make the following assumptions:

A1. $G[R]$ is a complete graph and, for any two terminals $u, v \in R$, c_{uv} is the cost of a minimum-cost u,v-path in G.

A2. For every Steiner vertex v and every vertex $u \in S_v \cup R$, uv is an edge of G, and c_{uv} is the cost of a minimum-cost u,v-path in G.

It is a well-known fact that these assumptions are w.l.o.g., i.e., any given instance can be transformed into an equivalent instance that satisfies these assumptions (e.g., see [36]). Note that b-quasi-bipartiteness is preserved by these assumptions.

Recall from Section 1.1 that a full component is a tree whose internal vertices are Steiner vertices and all of whose leaves are terminals. Also recall that a full component K is r-restricted if it contains at most r terminals. Further, the edge-set of any r-restricted Steiner tree T can be partitioned into r-restricted full components. From now on, let $r \geq 2$ be an arbitrary fixed constant. Define

$$\mathscr{K}_r := \{K \subseteq R : 2 \leq |K| \leq r \text{ and there exists a full component whose terminal set is } K\}.$$

We note that, for each $K \in \mathscr{K}_r$, we can determine a minimum-cost full component with terminal set K in polynomial time (e.g., by using the dynamic programming algorithm of Dreyfus and Wagner [11]). Thus, we can compute $\mathscr{K}_r$ in polynomial time as well.

For brevity we will abuse notation slightly and use $K \in \mathscr{K}_r$ interchangeably for a subset of the terminal set and for a particular min-cost full component spanning K. Given any r-restricted Steiner tree, we may assume that all of its full components are from $\mathscr{K}_r$, without increasing its cost.

For each full component K, we use $E(K)$ to denote its edges, $V(K)$ to denote its vertices (including Steiner vertices), and c_K to denote its cost. For a set $\mathscr{S}$ of full components we define $E(\mathscr{S}) := \cup_{K \in \mathscr{S}} E(K)$ and similarly $V(\mathscr{S}) := \cup_{K \in \mathscr{S}} V(K)$. By assumption A1 we may assume that the full component for a terminal pair is just the edge linking those terminals, and by assumption A2 we may assume that any Steiner node has degree at least 3. We will also assume that any two distinct full components $K_1, K_2 \in \mathscr{K}_r$ are edge disjoint and internally vertex disjoint. This assumption is without loss of generality as each Steiner vertex in G can be cloned a sufficient number of times to ensure this property. Finally, we redefine G to be $(V(\mathscr{K}_r), E(\mathscr{K}_r))$; as a result, the Steiner trees of the new graph correspond to the r-restricted Steiner trees of the original graph.

Let $\mathscr{K}_r(T)$ denote the set of all full components of a Steiner tree T. For an arbitrary subfamily $\mathscr{S}$ of the full components $\mathscr{K}_r$, our new LP uses the following canonical decomposition of a Steiner tree into elements of $E(\mathscr{S})$ and $\mathscr{K}_r \backslash \mathscr{S}$. The idea, as we will explain later, is to iteratively select a "good" set $\mathscr{S}$.

Definition 4. *If T is an r-restricted Steiner tree, its $\mathscr{S}$-*decomposition *is the pair*

$$(E(T) \cap E(\mathscr{S}), \mathscr{K}_r(T) \backslash \mathscr{S}).$$

Observe that after $\mathscr{S}$-decomposing a Steiner tree T we have

$$\sum_{e\in E(T)\cap E(\mathscr{S})} c_e + \sum_{K\in\mathscr{K}_r(T)\setminus\mathscr{S}} c_K = c(T).$$

We hence obtain a new higher-dimensional view of the Steiner tree polyhedron. Define

$$\mathtt{ST}^{\mathscr{S}}_{G,R} := \mathtt{conv}\{x\in\{0,1\}^{E(\mathscr{S})}\times\{0,1\}^{\mathscr{K}_r\setminus\mathscr{S}} \quad:\quad \exists T\in\mathtt{ST}_{G,R} \text{ s.t. } x \text{ is the incidence vector of the } \mathscr{S}\text{-decomposition of } T\}.$$

The following definitions are used to generalize Steiner partition inequalities to use full components. We use $\Pi^{\mathscr{S}}$ to denote the family of all partitions of $V(\mathscr{S})\cup R$.

Definition 5. *Let $\pi=\{V_1,\dots,V_p\}\in\Pi^{\mathscr{S}}$ be a partition of the set $R\cup V(\mathscr{S})$. The* rank contribution *of full component $K\in\mathscr{K}_r\setminus\mathscr{S}$ is defined as*

$$\mathtt{rc}^{\pi}_K := |\{i : K \text{ contains a terminal in } V_i\}| - 1.$$

The Steiner rank $\bar{r}(\pi)$ *of π is defined as*

$$\bar{r}(\pi) := \{\text{the number of parts of } \pi \text{ that contain terminals}\}.$$

We describe below a new LP relaxation ($\mathrm{P}^{\mathscr{S}}_{ST}$) of $\mathtt{ST}^{\mathscr{S}}_{G,R}$. The relaxation has a variable x_e for each $e\in E(\mathscr{S})$ and a variable x_K for each $K\in\mathscr{K}_r\setminus\mathscr{S}$. For a partition $\pi\in\Pi^{\mathscr{S}}$, we define $E_\pi(\mathscr{S})$ to be the edges of $\mathscr{S}$ whose endpoints lie in different parts of π, i.e., $E_\pi(\mathscr{S}) = E(\mathscr{S})\cap E_\pi$.

$$\begin{aligned}
\min \quad & \sum_{e\in E(\mathscr{S})} c_e\cdot x_e + \sum_{K\in\mathscr{K}_r\setminus\mathscr{S}} c_K\cdot x_K & & \qquad (\mathrm{P}^{\mathscr{S}}_{ST})\\
\text{s.t} \quad & \sum_{e\in E_\pi(\mathscr{S})} x_e + \sum_{K\in\mathscr{K}_r\setminus\mathscr{S}} \mathtt{rc}^{\pi}_K\cdot x_K \ge \bar{r}(\pi)-1 & \forall\pi\in\Pi^{\mathscr{S}} & \qquad (4)\\
& x_e, x_K \ge 0 & \forall e\in E(\mathscr{S}), K\in\mathscr{K}_r\setminus\mathscr{S} & \qquad (5)
\end{aligned}$$

Its LP dual has a variable y_π for each partition $\pi\in\Pi^{\mathscr{S}}$:

$$\begin{aligned}
\max \quad & \sum_{\pi\in\Pi^{\mathscr{S}}} (\bar{r}(\pi)-1)\cdot y_\pi & & \qquad (\mathrm{D}^{\mathscr{S}}_{ST})\\
\text{s.t} \quad & \sum_{\pi\in\Pi^{\mathscr{S}}: e\in E_\pi(\mathscr{S})} y_\pi \le c_e & \forall e\in E & \qquad (6)\\
& \sum_{\pi\in\Pi^{\mathscr{S}}} \mathtt{rc}^{\pi}_K\cdot y_\pi \le c_K & \forall K\in\mathscr{K}_r\setminus\mathscr{S} & \qquad (7)\\
& y_\pi \ge 0, & \forall\pi\in\Pi^{\mathscr{S}} & \qquad (8)
\end{aligned}$$

We conclude this section with a proof that the (primal) LP is indeed a relaxation of the convex hull of $\mathscr{S}$-decompositions for r-restricted Steiner trees. Obviously, constraints (5) hold whenever x is the incidence vector of the $\mathscr{S}$-decomposition of a Steiner tree.

Lemma 6. *The inequality* (4) *is valid for* $\mathtt{ST}^{\mathscr{S}}_{G,R}$.

Proof. Suppose, for the sake of contradiction, that (4) is not valid for $\mathtt{ST}^{\mathscr{S}}_{G,R}$ for this π. Then there must exist a feasible Steiner tree T with $\mathscr{S}$-decomposition $(E(T)\cap E(\mathscr{S}), \mathscr{K}_r(T)\setminus\mathscr{S})$ whose incidence vector $x \in \mathtt{ST}^{\mathscr{S}}_{G,R}$ violates (4) for some partition $\pi \in \Pi^{\mathscr{S}}$. Choose such a partition π with smallest rank.

Observe first that π must be a Steiner partition. Otherwise, there is a part V_1 of π that contains no terminals. Let V_2 be a part in π that contains terminals and obtain a new partition π' from π by merging V_1 and V_2. As V_1 contains no terminals, we clearly have $\mathtt{rc}^{\pi}_K = \mathtt{rc}^{\pi'}_K$ for all full components $K \in \mathscr{K}_r$. Also, the Steiner rank of π and π' is the same. As $e \in E_{\pi'}(\mathscr{S})$ implies that $e \in E_{\pi}(\mathscr{S})$, it follows that (4) is violated for π' as well and π' has smaller rank than π which contradicts our choice.

Suppose that $V(T) \subseteq R \cup V(\mathscr{S})$. This would mean that $\mathscr{K}_r(T)\setminus\mathscr{S} = \emptyset$ and in this case, Equation (3) implies that

$$\sum_{e \in E_\pi(\mathscr{S})} x_e \geq r(\pi) - 1.$$

Thus, inequality (4) holds for π and x which is a contradiction.

We may therefore assume that $\mathscr{K}_r(T)\setminus\mathscr{S}$ contains some full component $\bar{K}$. We obtain a new partition π' from π by merging those parts of π that contain terminals spanned by $\bar{K}$. The rank of this new partition is $r(\pi) - \mathtt{rc}^{\pi}_{\bar{K}}$. It follows from our choice of π that

$$\sum_{e \in E_{\pi'}(\mathscr{S})} x_e + \sum_{K \in \mathscr{K}_r\setminus\mathscr{S}} \mathtt{rc}^{\pi'}_K x_K \geq r(\pi') - 1 = r(\pi) - \mathtt{rc}^{\pi}_{\bar{K}} - 1.$$

Now note that $E_{\pi'}(\mathscr{S}) \subseteq E_\pi(\mathscr{S})$ and $\mathtt{rc}^{\pi'}_{\bar{K}} = 0$, and that $\mathtt{rc}^{\pi'}_K \leq \mathtt{rc}^{\pi}_K$ for all $K \in \mathscr{K}_r\setminus\mathscr{S}$. The above inequality therefore implies

$$\sum_{e \in E_\pi(\mathscr{S})} x_e + \sum_{K \in \mathscr{K}_r\setminus\mathscr{S}} \mathtt{rc}^{\pi}_K x_K \geq \sum_{e \in E_{\pi'}(\mathscr{S})} x_e + \sum_{K \in \mathscr{K}_r\setminus\mathscr{S}\setminus\{\bar{K}\}} \mathtt{rc}^{\pi'}_K x_K + \mathtt{rc}^{\pi}_{\bar{K}} \geq r(\pi) - \mathtt{rc}^{\pi}_{\bar{K}} - 1 + \mathtt{rc}^{\pi}_{\bar{K}}$$

which in turn proves that (4) holds for π and x. This contradiction completes the proof of the lemma. □

3 An iterated primal-dual algorithm for Steiner trees

As described in Section 2.2, $\mathtt{MST}(G,c)$ denotes a call to Kruskal's minimum-spanning tree algorithm on graph G with cost-function c. It returns a minimum-cost spanning tree T and an optimal feasible dual solution y for (D_{SP}). Let $\mathtt{mst}(G,c)$ denote the cost of $\mathtt{MST}(G,c)$. Since c is fixed, in the rest of the paper we omit c where possible for brevity. Let us also abuse notation and identify each set $\mathscr{S} \subset \mathscr{K}_r$ of full components with the graph $(V(\mathscr{S}), E(\mathscr{S}))$.

The main idea of the greedy algorithms in [34, 40, 41] is to find a set $\mathscr{S} \subset \mathscr{K}_r$ of full components such that $\mathtt{MST}(\mathscr{S})$ has small cost relative to $\mathtt{opt}_r$. Let $\binom{R}{2}$ denote the collection of all pairs of terminals. The algorithms all start with $\mathscr{S} = \binom{R}{2}$ and then grow $\mathscr{S}$, so for the rest of the paper we assume that $\binom{R}{2} \subseteq \mathscr{S}$; hence $E(G[R]) \subseteq E(\mathscr{S})$ and $R \subseteq V(\mathscr{S})$.

The reason that $\mathtt{MST}$ is useful in our primal-dual framework is that we can relate the dual program (D_{SP}) on graph $\mathscr{S}$ to the dual program ($\mathrm{D}^{\mathscr{S}}_{ST}$). Let y be the feasible dual returned by a call to $\mathtt{MST}(\mathscr{S})$. We treat y as a dual solution of ($\mathrm{D}^{\mathscr{S}}_{ST}$) by setting each y_K to zero; note that constraints (1) and (2) of (D_{SP}) imply that y also meets constraints (6) and (8) of ($\mathrm{D}^{\mathscr{S}}_{ST}$). If K is a full component such that (7) does not hold for y, we say that K is *violated* by y.

The primal-dual algorithm finds such a set $\mathscr{S}$ in an iterative fashion. Initially, $\mathscr{S}$ is equal to $\binom{R}{2}$. In each iteration, we compute a minimum-cost spanning tree T of the graph $\mathscr{S}$. The dual solution y corresponding to this tree is converted to a dual for ($\mathrm{D}^{\mathscr{S}}_{ST}$), and if y is feasible for ($\mathrm{D}^{\mathscr{S}}_{ST}$), we stop. Otherwise, we add a violated full component to $\mathscr{S}$ and continue. The algorithm clearly terminates (as $\mathscr{K}_r$ is finite) and at termination, it returns the final tree T as an approximately-optimal Steiner tree.

Algorithm 1 summarizes the above description. The greedy algorithms in [34, 40, 41] differ only in how K is selected in each iteration, i.e., in how the selection function $f_i : \mathscr{K}_r \to \mathbb{R}$ is defined (see also [19, §1.4] for a well-written comparison of these algorithms).

Algorithm 1 A general iterative primal-dual framework for Steiner trees.

```
Given: Undirected graph G = (V, E), non-negative costs c_e for all edges e ∈ E, constant r ≥ 2.
S^0 := (R choose 2), i := 0
repeat
    (T^i, y^i) := MST(S^i)
    if y^i is not feasible for (D_ST^{S^i}) then
        Choose a violated full component K^i ∈ K_r \ S^i such that f_i(K^i) is minimized
        S^{i+1} := S^i ∪ {K^i}
    end if
    i := i + 1
until y^{i-1} is feasible for (D_ST^{S^{i-1}})
Let p = i - 1 and return (T^p, y^p).
```

The following lemma is at the heart of our proof, and explains why our LP can be used to find cheap Steiner trees.

Lemma 7. *Let $(T, y) = \mathtt{MST}(\mathscr{S})$ and suppose that K is violated by y. Then adding K to $\mathscr{S}$ produces a cheaper spanning tree, i.e.,*

$$\mathtt{mst}(\mathscr{S} \cup \{K\}) < c(T).$$

Proof. Assume that $\mathtt{MST}(\mathscr{S})$ finishes at time τ^* and, once again, let π_τ be the partition maintained by Kruskal's algorithm at time $0 \le \tau \le \tau^*$.

Define $q = \mathtt{rc}_K^{\pi_0}$ to be the rank-contribution of K with respect to the initial partition. Clearly, $\mathtt{rc}_K^{\pi_{\tau^*}} = 0$ as all terminals are contained in the same connected component at time τ^*. Then there are edges $e_1, \dots, e_q \in T$ such that, for $1 \le i \le q$, the rank-contribution of K with respect to the partition maintained by Kruskal's algorithm drops from $q - i + 1$ to $q - i$ when edge e_i is added. Formally, for $1 \le i \le q$, let π_i and π_i' be the partition maintained by Kruskal's algorithm before and after adding edge e_i, then

$$\mathtt{rc}_K^{\pi_i} = \mathtt{rc}_K^{\pi_i'} + 1.$$

We denote the time of addition of edge e_i by τ_i for all i.

From the description of Kruskal's algorithm it follows that

$$\sum_{i=1}^{q} c_{e_i} = \sum_{i=1}^{q} \tau_i = \int_0^{\tau^*} \mathtt{rc}_K^{\pi_\tau} \, d\tau$$

and the right-hand side of this equality is equal to $\sum_{\pi \in \Pi^{\mathscr{S}}} \mathtt{rc}_K^{\pi} y_\pi$. The fact that constraint (7) is violated for K therefore implies that

$$c_{e_1} + \cdots + c_{e_q} > c_K.$$

Finally observe that $T \cup E(K) \setminus \{e_1, \dots, e_q\}$ is a spanning tree of $\mathscr{S} \cup \{K\}$ and its cost is smaller than that of T. ∎

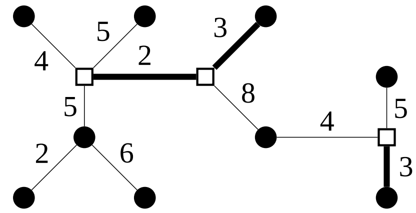

Figure 2: The figure shows the Steiner tree instance from Figure 1 with costs on the edges. The loss of the Steiner tree in this figure is shown in thick edges. Its cost is 8.

3.1 Cutting losses: the $\mathtt{RZ}$ selection function

A potential weak point in Algorithm 1 is that once a full component is added to $\mathscr{S}$, it is never removed. On the other hand, if some cheap subgraph H connects all Steiner vertices of $\mathscr{S}$ to terminals, then adding H to any Steiner tree gives us a tree that spans $V(\mathscr{S})$, i.e., we have so far *lost* at most $c(H)$ in the final answer. This leads to the concept of the *loss* of a Steiner tree which was first introduced by Karpinski and Zelikovsky in [22].

Definition 8. *Let $G' = (V', E')$ be a subgraph of G. The* loss $\mathtt{L}(G')$ *is a minimum-cost set $E'' \subseteq E'$ such that every connected component of (V', E'') contains a terminal. Let $\mathtt{l}(G')$ denote the cost of $\mathtt{L}(G')$.*

See Figure 2 for an example of the loss of a graph. The above discussion amounts to saying that $\min\{\mathtt{mst}(\mathscr{S}') \mid \mathscr{S}' \supseteq \mathscr{S}\} \leq \mathtt{opt}_r + \mathtt{l}(\mathscr{S})$. Consequently, our selection function f_i in step 6 of the algorithm should try to keep the loss small. The following fact holds because full components in $\mathscr{K}_r$ meet only at terminals.

Fact 9. *If $\mathscr{S} \subseteq \mathscr{K}_r$, then $\mathtt{L}(\mathscr{S}) = \cup_{K \in \mathscr{S}} \mathtt{L}(K)$ and so $\mathtt{l}(\mathscr{S}) = \sum_{K \in \mathscr{S}} \mathtt{l}(K)$.*

For a set $\mathscr{S}$ of full components, where y is the dual solution returned by $\mathtt{MST}(\mathscr{S})$, define

$$\overline{\mathtt{mst}}(\mathscr{S}) := \sum_{\pi \in \Pi^{\mathscr{S}}} (\overline{r}(\pi) - 1) y_\pi. \tag{9}$$

If y is feasible for $(\mathrm{D}^{\mathscr{S}}_{ST})$ then by weak LP duality, $\overline{\mathtt{mst}}(\mathscr{S})$ provides a lower bound on $\mathtt{opt}_r$. If y is infeasible for $(\mathrm{D}^{\mathscr{S}}_{ST})$, then which full component should we add? Robins and Zelikovsky propose minimizing the ratio of the change in upper bound to the change in potential lower bound (9). Their selection function f_i is defined by

$$f_i(K) := \frac{\mathtt{l}(K)}{\overline{\mathtt{mst}}(\mathscr{S}^i) - \overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K\})} = \frac{\mathtt{l}(\mathscr{S}^i \cup \{K\}) - \mathtt{l}(\mathscr{S}^i)}{\overline{\mathtt{mst}}(\mathscr{S}^i) - \overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K\})}, \tag{10}$$

where the equality uses Fact 9.

4 Analysis

Fix an optimum r-Steiner tree T^*. There are several steps in proving the performance guarantee of Robins and Zelikovsky's algorithm, and they are encapsulated in the following result, whose complete proof appears in Section 6.

Lemma 10. *The cost of the tree T^p returned by Algorithm 1 is at most*

$$\mathtt{opt}_r + \mathtt{l}(T^*) \cdot \ln\left(1 + \frac{\overline{\mathtt{mst}}(G[R], c) - \mathtt{opt}_r}{\mathtt{l}(T^*)}\right).$$

The main observation in the proof of the above lemma can be summarized as follows: from the discussion in Section 2, we know that the tree T^p returned by Algorithm 1 has cost

$$\mathtt{mst}(\mathscr{S}^p) = \sum_{\pi \in \Pi^{\mathscr{S}^p}} (r(\pi) - 1) y^p_\pi$$

and the corresponding lower-bound on $\mathtt{opt}_r$ returned by the algorithm is

$$\overline{\mathtt{mst}}(\mathscr{S}^p) = \sum_{\pi \in \Pi^{\mathscr{S}^p}} (\overline{r}(\pi) - 1) y^p_\pi.$$

We know that $\overline{\mathtt{mst}}(\mathscr{S}^p) \leq \mathtt{opt}_r$ but how large is the difference between $\mathtt{mst}(\mathscr{S}^p)$ and $\overline{\mathtt{mst}}(\mathscr{S}^p)$? We show that the difference

$$\sum_{\pi \in \Pi^{\mathscr{S}^p}} (r(\pi) - \overline{r}(\pi)) y^p_\pi$$

is exactly equal to the loss $\mathtt{l}(T^p)$ of tree T^p. We then bound the loss of each selected full component K^i, and putting everything together finally yields Lemma 10.

The following lemma states the performance guarantee of Moore's minimum-spanning tree heuristic as a function of the optimum loss and the maximum cardinality b of any Steiner neighborhood in G.

Lemma 11. *Fix an arbitrary optimum r-restricted Steiner tree T^*. Given an undirected, b-quasi-bipartite graph $G = (V, E)$, a set of terminals $R \subseteq V$, and non-negative costs c_e for all $e \in E$, we have*

$$\mathtt{mst}(G[R], c) \leq 2\mathtt{opt}_r - \frac{2}{b}\mathtt{l}(T^*)$$

for any $b \geq 1$.

Proof. Recall that $\mathscr{K}_r(T^*)$ is the set of full components of tree T^*. Now consider a full component $K \in \mathscr{K}_r(T^*)$. We will now show that there is a minimum-cost spanning tree of $G[K]$ whose cost is at most $2c_K - \frac{2}{b}\mathtt{l}(K)$. By repeating this argument for all full components $K \in \mathscr{K}_r(T^*)$, adding the resulting bounds, and applying Fact 9, we obtain the lemma.

For terminals $r, s \in K$, let P_{rs} denote the unique r,s-path in K. Pick $u, v \in K$ such that $c(P_{uv})$ is maximal. Define the *diameter* $\Delta(K) := c(P_{uv})$. Do a depth-first search traversal of K starting in u and ending in v. The resulting walk in K traverses each edge not on P_{uv} twice while each edge on P_{uv} is traversed once. Hence the walk has cost $2c_K - \Delta(K)$. Using standard short-cutting arguments it follows that the minimum-cost spanning tree of $G[K]$ has cost at most

$$2c_K - \Delta(K) \tag{11}$$

as well.

Each Steiner vertex $s \in V(K) \backslash R$ can connect to some terminal $v \in K$ at cost at most $\frac{\Delta(K)}{2}$. Hence, the cost $\mathtt{l}(K)$ of the loss of K is at most $b\frac{\Delta(K)}{2}$. In other words we have $\Delta(K) \geq \frac{2}{b}\mathtt{l}(K)$. Plugging this into (11) yields the lemma. □

For small values of b we can obtain additional improvements via case analysis.

Lemma 12. *Suppose $b \in \{3, 4\}$. Fix an arbitrary optimum r-restricted Steiner tree T^*. Given an undirected, b-quasi-bipartite graph $G = (V, E)$, a set of terminals $R \subseteq V$, and non-negative costs c_e for all $e \in E$, we have*

$$\mathtt{mst}(G[R], c) \leq 2\mathtt{opt}_r - \mathtt{l}(T^*).$$

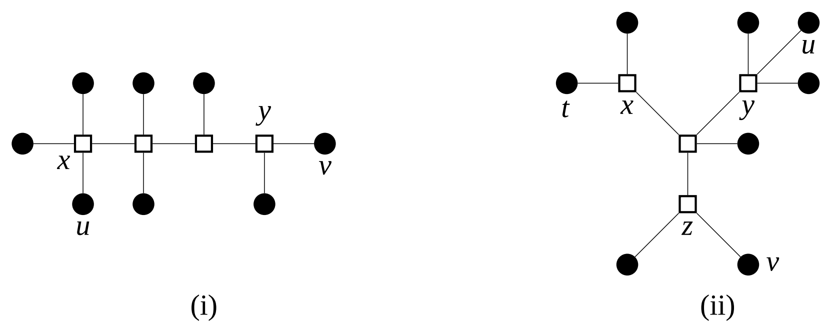

Figure 3: The figure shows the two types of full components when $b \leq 4$. On the left is a full component where the Steiner nodes form a path, and on the right is a full component where the Steiner nodes form a star with 3 tips.

Proof. As in the proof of Lemma 11 it suffices to prove that, for each full component $K \in \mathscr{K}_r(T^*)$, there is a minimum-cost spanning tree of $G[K]$ whose cost is at most $2c_K - \mathtt{l}(K)$, for then we can add the bound over all such K to get the desired result. For terminals $r, s \in K$, let P_{rs} again denote the unique r,s-path in K.

Notice that the Steiner nodes (there are at most b of them) in the full component K either form a path, or else there are 4 of them and they form a star.

Case 1: the Steiner nodes in K form a path. Let x and y be the Steiner nodes on the ends of this path. Let u (resp. v) be any terminal neighbour of x (resp. y); see Figure 3(i) for an example. Perform a depth-first search in K starting from u and ending at v; the cost of this search is $2c_K - c(P_{uv})$. By standard short-cutting arguments it follows that $2c_K - c(P_{uv})$ is an upper bound on $\mathtt{mst}(G[K])$. On the other hand, since $P_{uv} \setminus \{ux\}$ is a candidate for the loss of K, we know that $\mathtt{l}(K) \leq c(P_{uv} \setminus \{ux\}) \leq c(P_{uv})$. Therefore we obtain

$$\mathtt{mst}(G[K]) \leq 2c_K - c(P_{uv}) \leq 2c_K - \mathtt{l}(K). \tag{12}$$

Case 2: the Steiner nodes in K form a star. Let the tips of the star be x, y, z and let t, u, v be any terminal neighbours of x, y, z respectively; see Figure 3(ii) for an example. Without loss of generality, we may assume that $c_{xt} \leq c_{yu} \leq c_{zv}$. As before, a depth-first search in K starting from u and ending at v has cost $2c_K - c(P_{uv})$ and this is an upper bound on $\mathtt{mst}(G[K])$. On the other hand, $P_{uv} \setminus \{yu\} \cup \{xt\}$ is a candidate for the loss of K and so $\mathtt{l}(K) \leq c(P_{uv}) - c_{yu} + c_{xt} \leq c(P_{uv})$. We hence obtain Equation (12) as in the previous case. $\square$

We are ready to prove our main theorem. We restate it using the notation introduced in the last two sections.

Theorem 1. *Given an undirected, b-quasi-bipartite graph $G = (V, E)$, terminals $R \subseteq V$, and a fixed constant $r \geq 2$, Algorithm 1 returns a feasible Steiner tree T^p with*

$$c(T^p) \leq \begin{cases} 1.279 \cdot \mathtt{opt}_r & : \quad b = 1 \\ (1 + 1/e) \cdot \mathtt{opt}_r & : \quad b \in \{2,3,4\} \\ \left(1 + \frac{1}{2}\ln\left(3 - \frac{2}{b}\right)\right) \mathtt{opt}_r & : \quad b \geq 5. \end{cases}$$

Proof. Using Lemma 10 we see that

$$\begin{aligned} c(T^p) &\leq \mathtt{opt}_r + \mathtt{l}(T^*)\cdot \ln\left(1+\frac{\overline{\mathtt{mst}}(G[R],c)-\mathtt{opt}_r}{\mathtt{l}(T^*)}\right) \\ &= \mathtt{opt}_r + \mathtt{l}(T^*)\cdot \ln\left(1+\frac{\mathtt{mst}(G[R],c)-\mathtt{opt}_r}{\mathtt{l}(T^*)}\right). \end{aligned} \tag{13}$$

The second equality above holds because $G[R]$ has no Steiner vertices. Applying the bound on $\mathtt{mst}(G[R],c)$ from Lemma 11 yields

$$c(T^p) \leq \mathtt{opt}_r \cdot \left[1+\frac{\mathtt{l}(T^*)}{\mathtt{opt}_r}\cdot \ln\left(1-\frac{2}{b}+\frac{\mathtt{opt}_r}{\mathtt{l}(T^*)}\right)\right]. \tag{14}$$

Karpinski and Zelikovsky [22] show that $\mathtt{l}(T^*) \leq \frac{1}{2}\mathtt{opt}_r$. We can therefore obtain an upper-bound on the right-hand side of (14) by bounding the maximum value of function $x\ln(1-2/b+1/x)$ for $x \in [0,1/2]$. We branch into cases:

$b=1$: The maximum of $x\ln(1/x-1)$ for $x\in[0,1/2]$ is attained for $x\approx 0.2178$. Hence, $x\ln(1/x-1)\leq 0.279$ for $x\in[0,1/2]$.

$b=2$: The maximum of $x\ln(1/x)$ is attained for $x=1/e$ and hence $x\ln(1/x)\leq 1/e$ for $x\in[0,1/2]$.

$b\in\{3,4\}$: We use Equation (13) together with Lemma 12 in place of Lemma 11; the subsequent analysis and result are the same as in the previous case.

$b\geq 5$: The function $x\ln(1-2/b+1/x)$ is increasing in x and its maximum is attained for $x=1/2$. Thus, $x\ln(1-2/b+1/x)\leq \frac{1}{2}\ln(3-2/b)$ for $x\in[0,1/2]$.

The three cases above conclude the proof of the theorem. □

5 Properties of $(\mathrm{P}_{ST}^{\mathscr{S}})$

In this section, we first prove that the linear program $(\mathrm{P}_{ST}^{\mathscr{S}})$ is gradually weakened as the algorithm progresses (i.e., as more full components are added to $\mathscr{S}$). Then we describe bounds on the integrality gap of the new LP, and its strength compared to other LPs for the Steiner tree problem.

Lemma 13. *If $\mathscr{S}\subset\mathscr{S}'$, then the integrality gap of $(\mathrm{P}_{ST}^{\mathscr{S}})$ is at most the integrality gap of $(\mathrm{P}_{ST}^{\mathscr{S}'})$.*

Proof. We consider only the case where $\mathscr{S}'=\mathscr{S}\cup\{J\}$ for some full component J; the general case then follows by induction on $|\mathscr{S}'\backslash\mathscr{S}|$.

Let x be any feasible primal point for $(\mathrm{P}_{ST}^{\mathscr{S}})$ and define the *extension* x' of x to be a primal point of $(\mathrm{P}_{ST}^{\mathscr{S}'})$, with $x'_e = x_J$ for all $e\in E(J)$ and $x'_Z = x_Z$ for all $Z\in(\mathscr{K}_r\backslash\mathscr{S}')\cup E(\mathscr{S})$. We claim that x' is feasible for $(\mathrm{P}_{ST}^{\mathscr{S}'})$. Since x and x' have the same objective value, this will prove Lemma 13.

It is clear that x' satisfies constraints (5), so now let us show that x' satisfies the partition inequality (4) in $(\mathrm{P}_{ST}^{\mathscr{S}'})$. Fix an arbitrary partition π' of $V(\mathscr{S}')$, and let π be the restriction of π' to $V(\mathscr{S})$. We get

$$\sum_{e\in E_{\pi'}(\mathscr{S}')} x'_e + \sum_{K\in\mathscr{K}_r\backslash\mathscr{S}'} \mathtt{rc}_K^{\pi'} x'_K = \left(\sum_{e\in E_{\pi}(\mathscr{S})} x_e + \sum_{K\in\mathscr{K}_r\backslash\mathscr{S}} \mathtt{rc}_K^{\pi} x_K\right) + |E_{\pi'}\cap E(J)|x_J - \mathtt{rc}_J^{\pi} x_J. \tag{15}$$

Now J spans at least $\mathtt{rc}_J^{\pi}+1$ parts of π', and it follows that $|E_{\pi'}\cap E(J)|\geq \mathtt{rc}_J^{\pi}$. Hence, using Equation (15), the fact that x satisfies constraint (4) for π, and the fact that $\overline{r}(\pi)=\overline{r}(\pi')$, we have

$$\sum_{e\in E_{\pi'}(\mathscr{S}')} x'_e + \sum_{K\in\mathscr{K}_r\backslash\mathscr{S}'} \mathtt{rc}_K^{\pi'} x'_K \geq \sum_{e\in E_{\pi}(\mathscr{S})} x_e + \sum_{K\in\mathscr{K}_r\backslash\mathscr{S}} \mathtt{rc}_K^{\pi} x_K \geq \overline{r}(\pi)-1 = \overline{r}(\pi')-1.$$

So x' satisfies (4) for π'. □

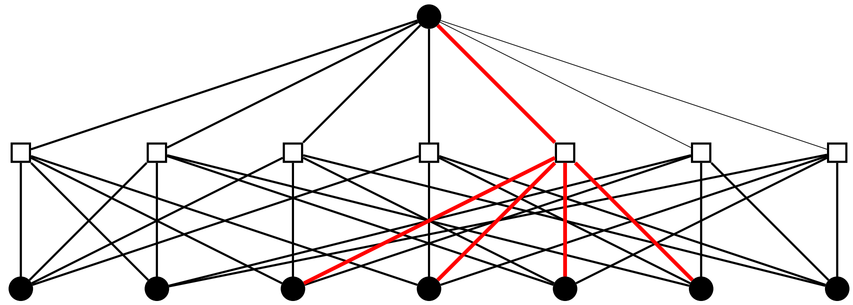

Figure 4: Skutella's example, which shows that the bidirected cut formulation and our new formulation both have a gap of at least $\frac{8}{7}$. The shaded edges denote one of the quasi-bipartite full components on 5 terminals.

In 1997, Warme [37] introduced a new linear program for the Steiner tree problem. He observed (as did the authors of [30] in the same year) that full components allow a reduction from the Steiner tree problem to the *spanning-tree-in-hypergraph* problem. He also gave an LP relaxation for spanning trees in hypergraphs. That LP turns out to be exactly as strong as our own LP; see [24, Corollary 3.19] for a proof. Now, Polzin et al. [29] proved that Warme's relaxation is stronger than the bidirected cut relaxation, and Goemans [15] proved that the (graph) Steiner partition inequalities are valid for the bidirected cut formulation. Hence, as stated previously, using full components as in $(\mathrm{P}^{\mathscr{S}}_{ST})$ strengthens the Steiner partition inequalities.

5.1 A lower bound on the integrality gap of $(\mathrm{P}^{\emptyset}_{ST})$

Note that when $\mathscr{S} = \binom{R}{2}$, $(\mathrm{P}^{\emptyset}_{ST})$ and $(\mathrm{P}^{\mathscr{S}}_{ST})$ are equivalent LPs: for each terminal-terminal edge uv, the full component variable $x_{\{u,v\}}$ of the former corresponds to the edge variable x_{uv} of the latter. Hence although we consider the simpler LP $(\mathrm{P}^{\emptyset}_{ST})$ in this section, the results apply also to the LP used in the first iteration of $\mathtt{RZ}$.

Goemans [1] gave a family of graphs upon which, in the limit, the integrality gap of the bidirected cut relaxation is $\frac{8}{7}$. Interestingly, it can be shown that once you preprocess these graphs as described in Section 2.3, the gap completely disappears. Here we describe another example, due to Skutella [35]. It shows not only that the gap of the bidirected cut relaxation is at least $\frac{8}{7}$, but that the gap of our new formulation (including preprocessing) is at least $\frac{8}{7}$. The example is quasi-bipartite.

The Fano design is a well-known finite geometry consisting of 7 *points* and 7 *lines*, such that every point is on 3 lines, every line contains 3 points, any two lines meet in a unique point, and any two points lie on a unique common line. We construct Skutella's example by creating a bipartite graph, with one side consisting of one node n_p for each point p of the Fano design, and the other side consisting of one node n_ℓ for each line ℓ of the Fano design. Define n_p and n_ℓ to be adjacent in our graph if and only if p does *not* lie on ℓ. Then it is easy to see this graph is 4-regular, and that given any two nodes n_1, n_2 from one side, there is a node from the other side that is adjacent to neither n_1 nor n_2. Let one side be terminals, the other side be Steiner nodes, and then attach one additional terminal to all the Steiner nodes. We illustrate the resulting graph in Figure 4.

Each Steiner node is in a unique 5-terminal quasi-bipartite full component. There are 7 such full components. Denote the family of these 7 full components by $\mathscr{C}$.

Claim 14. *Let $x^*_K = \frac{1}{4}$ for each $K \in \mathscr{C}$, and $x^*_K = 0$ otherwise. Then x^* is feasible for $(\mathrm{P}^{\emptyset}_{ST})$.*

Proof. It is immediate that x^* satisfies constraints (5). It remains only to show that x^* meets constraint (4). Let π be an arbitrary partition, with parts $\pi_0, \ldots, \pi_m$ such that π_0 contains the extra "top" terminal. If we can show that $\sum_K x^*_K \mathtt{rc}^{\pi}_K \geq m$ then we will be done, since π was arbitrary. For each $i = 1, \ldots, m$, let r_i be any terminal in π_i. Note that each r_i lies in exactly 4 full components from $\mathscr{C}$. Furthermore, every full component

$K \in \mathscr{C}$ satisfies $\mathtt{rc}_K^\pi \geq |K \cap \{r_1, \ldots, r_m\}|$, since that full component meets π_0 as well as each part π_j such that $r_j \in K$. Hence

$$\sum_K x_K^* \mathtt{rc}_K^\pi = \frac{1}{4} \sum_{K \in \mathscr{C}} \mathtt{rc}_K^\pi \geq \frac{1}{4} \sum_{K \in \mathscr{C}} \#\{j : r_j \in K\} = \frac{1}{4} \sum_{j=1}^{m} \#\{K \in \mathscr{C} : r_j \in K\} = \frac{1}{4} \cdot m \cdot 4 = m. \qquad \square$$

The objective value of x^* is $\frac{35}{4}$, but the optimal integral solution to the LP is 10, since at least 3 Steiner nodes need to be included. Hence, the gap of our new LP is no better than $\frac{10}{35/4} = \frac{8}{7}$.

5.2 A gap upper bound for b-quasi-bipartite instances

In [32] Rajagopalan and Vazirani show that the bidirected cut relaxation has a gap of at most $\frac{3}{2}$, if the graph is quasi-bipartite. Since $(\mathrm{P}_{ST}^{\emptyset})$ is stronger than the bidirected cut relaxation its gap is also at most $\frac{3}{2}$ for such graphs. We are able to generalize this result as follows.

Theorem 2. *On b-quasi-bipartite graphs,* $(\mathrm{P}_{ST}^{\emptyset})$ *has an integrality gap between* $\frac{8}{7}$ *and* $\frac{2b+1}{b+1}$ *in the worst case.*

Proof. The lower bound comes from Section 5.1. We assume G is b-quasi-bipartite, we let T^* be an optimal Steiner tree, and we let $\mathscr{S}^*$ be its set of full components. Since T^* is a minimum spanning tree for $\mathscr{S}^*$, there is a corresponding feasible dual y for (D_{SP}). When we convert y to a dual for $(\mathrm{D}_{ST}^{\mathscr{S}^*})$, we claim that y is feasible: indeed, by Lemma 7 a violated full component could be used to improve the solution, but T^* is already optimal. The next lemma is the cornerstone of our proof.

Lemma 15. *Let π be a partition of $V(\mathscr{S}^*)$ with $y_\pi > 0$. Then* $(\overline{r}(\pi) - 1) \geq \frac{b+1}{2b+1}(r(\pi) - 1)$.

Proof. For each part π_i of π, let us identify all of the nodes of π_i into a single pseudonode v_i. We may assume by Theorem 3 that each $T^*[\pi_i]$ is connected, hence this identification process yields a tree T'. Let us say that v_i is *Steiner* if and only if all nodes of π_i are Steiner. Note that T' has $r(\pi)$ pseudonodes and $r(\pi) - \overline{r}(\pi)$ of these pseudonodes are Steiner. The full components of T' are defined analogously to the full components of a Steiner tree.

Consider any full component K' of T' and let K' contain exactly s Steiner pseudonodes. It is straightforward to see that $s \leq b$. Each Steiner pseudonode in K' has degree at least 3 by Assumptions A1 and A2, and at most $s-1$ edges of K' join Steiner vertices to other Steiner vertices. Hence K' has at least $3s - (s-1) = 2s+1$ edges, and so

$$|E(K')| \geq \frac{2s+1}{s} \cdot s \geq \frac{2b+1}{b} \cdot s.$$

Now summing over all full components K', we obtain

$$|E(T')| \geq \frac{2b+1}{b} \cdot \#\{\text{Steiner pseudonodes of } T'\}.$$

But $|E(T')| = r(\pi) - 1$ and T' has $r(\pi) - \overline{r}(\pi)$ Steiner pseudonodes, therefore

$$r(\pi) - 1 \geq \frac{2b+1}{b}((r(\pi)-1) - (\overline{r}(\pi) - 1)) \quad \Rightarrow \quad \frac{2b+1}{b}(\overline{r}(\pi) - 1) \geq \frac{b+1}{b}(r(\pi) - 1).$$

This proves what we wanted to show. $\square$

It follows that the objective value of y in $(\mathrm{D}_{ST}^{\mathscr{S}^*})$ is

$$\sum_{\pi \in \Pi^{\mathscr{S}}} (\overline{r}(\pi) - 1) y_\pi \geq \sum_{\pi \in \Pi^{\mathscr{S}}} \frac{b+1}{2b+1} (\overline{r}(\pi) - 1) y_\pi = \frac{b+1}{2b+1} c(T^*)$$

and since T^* is an optimum integer solution of $(\mathrm{P}_{ST}^{\mathscr{S}^*})$, it follows that the integrality gap of $(\mathrm{P}_{ST}^{\mathscr{S}^*})$ is at most $\frac{b+1}{2b+1}$. Then, finally, by applying Lemma 13 to $(\mathrm{P}_{ST}^{\emptyset})$ and $(\mathrm{P}_{ST}^{\mathscr{S}^*})$ we obtain Theorem 2. $\square$

6 Proof of Lemma 10

In this section we present a proof of Lemma 10. The methodology follows that proposed by Gröpl et al. [19]. In fact, many of the proofs below essentially correspond to those presented in [19] with two exceptions: we correct a small error near the end, and we present a new proof of the ubiquitous *contraction lemma*.

We remind the reader of our standing assumption that $\mathscr{S} \supseteq \binom{R}{2}$. We first relate the cost of a minimum-cost spanning tree of $\mathscr{S}$ for some set $\mathscr{S}$ of full components to the (potential) lower-bound $\overline{\mathtt{mst}}(\mathscr{S})$ on $\mathtt{opt}_r$ that it provides. For ease of presentation in the analysis, we will assume from now on that the costs of all edges in E are pairwise different. This assumption is easily seen to be w.l.o.g. (e.g., one could define an order on the edges in E and use it to break ties). We omit the proof of the following easy fact.

Fact 16. *If T is a minimum-cost spanning tree of $\mathscr{S}$ then $\mathtt{l}(T) = \mathtt{l}(\mathscr{S})$.*

Lemma 17. *For any set $\mathscr{S} \subseteq \mathscr{K}_r$ of full components,*

$$\mathtt{mst}(\mathscr{S}) = \overline{\mathtt{mst}}(\mathscr{S}) + \mathtt{l}(\mathscr{S}).$$

Proof. We use the notation from Section 2: τ^* is the finishing time of Kruskal's algorithm, $G_\tau = (V, E_\tau)$ is the forest maintained at time τ, and π_τ is the partition induced by the connected components of G_τ. Let (T,y) denote the tree-dual pair returned by $\mathtt{MST}$.

From Theorem 3 we know that there exists a feasible dual solution y to (D_{SP}) for graph $\mathscr{S}$ such that

$$c(T) = \sum_{\pi \in \Pi^{\mathscr{S}}} (r(\pi) - 1) y_\pi = \int_0^{\tau^*} (r(\pi_\tau) - 1) d\tau.$$

In the following let $\mathscr{R}_\tau$ be the set of those connected components of E_τ that contain terminal vertices.

Claim 18. *For all $0 \le \tau \le \tau^*$, each connected component of $E_\tau \cup \mathtt{L}(T)$ contains exactly one connected component of $\mathscr{R}_\tau$.*

Proof. Let u and v be terminals in distinct connected components of G_τ and let P_{uv} be the unique u,v-path in T. Assume for the sake of contradiction that P_{uv} is contained in $E_\tau \cup \mathtt{L}(T)$.

Let $\bar{e}$ be the unique edge of maximum cost on path P_{uv}. Recall from Section 2 that Kruskal's algorithm adds edges to the partial spanning tree in order of non-decreasing cost. Thus, edge $\bar{e}$ is added last among all edges on P_{uv}. As u and v are in different connected components of G_τ, it therefore follows that $\bar{e} \notin E_\tau$. The loss of T is a minimum-cost forest in T that connects all Steiner vertices to terminals. Thus, the unique edge of maximum cost on P_{uv} cannot be in $\mathtt{L}(T)$.

It follows that $\bar{e} \notin E_\tau \cup \mathtt{L}(T)$ and this contradicts our assumption that $P_{uv} \subseteq E_\tau \cup \mathtt{L}(T)$. ☐

For each time $0 \le \tau \le \tau^*$, define $\bar{\pi}_\tau$ as the Steiner partition corresponding to the connected components of $G_\tau \cup \mathtt{L}(T)$. From Theorem 3 we know that

$$\mathtt{l}(T) = \sum_{e \in \mathtt{L}(T)} c_e = \sum_{e \in \mathtt{L}(T)} \sum_{\pi : e \in E_\pi} y_\pi = \int_0^{\tau^*} |E_{\pi_\tau} \cap \mathtt{L}(T)| d\tau$$

where, as before, E_{π_τ} is the set of edges in E that have endpoints in different parts of π_τ.

The number of edges in $|E_{\pi_\tau} \cap \mathtt{L}(T)|$ is exactly the rank-difference between π_τ and $\bar{\pi}_\tau$ and hence

$$\mathtt{l}(T) = \int_0^{\tau^*} (r(\pi_\tau) - r(\bar{\pi}_\tau)) d\tau.$$

Claim 18 implies that $r(\overline{\pi}_\tau) = \overline{r}(\pi_\tau)$ for all $0 \le \tau \le \tau^*$ and hence

$$\overline{\mathtt{mst}}(\mathscr{S}) + \mathtt{l}(T) = \int_0^{\tau^*} (\overline{r}(\pi) - 1) d\tau + \int_0^{\tau^*} (r(\pi_\tau) - \overline{r}(\pi_\tau)) d\tau = \int_0^{\tau^*} (r(\pi) - 1) d\tau = c(T).$$

Applying Fact 16 and the equality $c(T) = \mathtt{mst}(\mathscr{S})$, we are done. □

We obtain the following immediate corollary:

Corollary 19. *In iteration i of Algorithm 1, adding full component $K \in \mathscr{K}_r$ to $\mathscr{S}$ reduces the cost of $\mathtt{mst}(\mathscr{S})$ if and only if $f_i(K) < 1$.*

Proof. By applying Lemma 17 we see that

$$\mathtt{mst}(\mathscr{S}^i) - \mathtt{mst}(\mathscr{S}^i \cup \{K\}) = \overline{\mathtt{mst}}(\mathscr{S}^i) + \mathtt{l}(\mathscr{S}^i) - \overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K\}) - \mathtt{l}(\mathscr{S}^i \cup \{K\}).$$

Whereas the left-hand side is positive iff adding K to $\mathscr{S}^i$ causes a reduction in $\mathtt{mst}$, the right-hand side is positive iff $f_i(K) < 1$, due to the definition of f_i. □

Using Lemma 7 and Corollary 19, we obtain the following.

Corollary 20. *For all* $1 \le i \le p$, $f_i(K^i) < 1$.

Fix an optimum r-Steiner tree T^*. The next two lemmas give bounds that are needed to analyze $\mathtt{RZ}$'s greedy strategy. Informally, the first says that $\overline{\mathtt{mst}}$ is non-increasing, while the second says that $\overline{\mathtt{mst}}$ is submodular.

Lemma 21. *If $\mathscr{S} \subseteq \mathscr{S}' \subseteq \mathscr{K}_r$, then $\overline{\mathtt{mst}}(\mathscr{S}') \le \overline{\mathtt{mst}}(\mathscr{S})$.*

Proof. Using Lemma 17 and Fact 9 we see

$$\overline{\mathtt{mst}}(\mathscr{S}) - \overline{\mathtt{mst}}(\mathscr{S}') = \mathtt{mst}(\mathscr{S}) + \mathtt{l}(\mathscr{S}' \backslash \mathscr{S}) - \mathtt{mst}(\mathscr{S}').$$

However, the right hand side of the above equation is non-negative, as $\mathtt{MST}(\mathscr{S}) \cup \mathtt{L}(\mathscr{S}' \backslash \mathscr{S})$ is a spanning tree of $\mathscr{S}'$. Lemma 21 then follows. □

Lemma 22 (Contraction Lemma)**.** *Let $\mathscr{R}^0, \mathscr{R}^1, \mathscr{R}^2 \subset \mathscr{K}_r$ be disjoint collections of full components with $\binom{R}{2} \subseteq \mathscr{R}^0$. Then*

$$\overline{\mathtt{mst}}(\mathscr{R}^0) - \overline{\mathtt{mst}}(\mathscr{R}^0 \cup \mathscr{R}^2) \ge \overline{\mathtt{mst}}(\mathscr{R}^0 \cup \mathscr{R}^1) - \overline{\mathtt{mst}}(\mathscr{R}^0 \cup \mathscr{R}^1 \cup \mathscr{R}^2).$$

Proof. The statement to be proved is equivalent to

$$\mathtt{mst}(\mathscr{R}^0) - \mathtt{mst}(\mathscr{R}^0 \cup \mathscr{R}^2) \ge \mathtt{mst}(\mathscr{R}^0 \cup \mathscr{R}^1) - \mathtt{mst}(\mathscr{R}^0 \cup \mathscr{R}^1 \cup \mathscr{R}^2), \tag{16}$$

due to Lemma 17 and Fact 9. For a graph H, define the *rank* $r(H)$ of H as the number of edges in a maximal forest of H:

$$r(H) = |V(H)| - \#\text{ connected components of } H.$$

For a graph H, let $H_{\le x}$ denote the subgraph of H consisting of those edges of weight at most x. By considering Kruskal's algorithm, for any graph H having nonnegative edge costs, we see that

$$\mathtt{mst}(H) = \sum_{i=1}^{r(H)} \min\{x \mid r(H_{\le x}) \ge i\} = \int_0^\infty \big(r(H) - r(H_{\le x})\big)\, dx. \tag{17}$$

Note that the integral is proper since the integrand is 0 for x larger than $\max\{c_e : e \in E(H)\}$.

Here is the crux: r is the rank function for a (graphic) matroid and is therefore submodular over the addition of disjoint edge sets. Since the $\mathscr{R}^i_{<x}$ are pairwise disjoint, for every x, this submodularity implies that

$$-r\left(\mathscr{R}^0_{\le x}\right) + r\left(\mathscr{R}^0_{\le x} \cup \mathscr{R}^2_{\le x}\right) \ge -r\left(\mathscr{R}^0_{\le x} \cup \mathscr{R}^1_{\le x}\right) + r\left(\mathscr{R}^0_{\le x} \cup \mathscr{R}^1_{\le x} \cup \mathscr{R}^2_{\le x}\right). \tag{18}$$

Notice also that

$$r(\mathscr{R}^0) - r(\mathscr{R}^0 \cup \mathscr{R}^2) = r(\mathscr{R}^0 \cup \mathscr{R}^1) - r(\mathscr{R}^0 \cup \mathscr{R}^1 \cup \mathscr{R}^2) \tag{19}$$

since both sides are equal to the number of Steiner vertices in $\mathscr{R}^2$, times -1.

Finally, we add Equation (18) to Equation (19) and integrate along x. Since $(\mathscr{R}^0 \cup \mathscr{R}^2)_{\le x} = \mathscr{R}^0_{<x} \cup \mathscr{R}^2_{\le x}$ etc. we get

$$\begin{aligned}
&\int_0^\infty \left(r(\mathscr{R}^0) - r(\mathscr{R}^0_{\le x})\right) dx - \int_0^\infty \Big(r(\mathscr{R}^0 \cup \mathscr{R}^2) - r\left((\mathscr{R}^0 \cup \mathscr{R}^2)_{\le x}\right)\Big) dx \\
&\ge \int_0^\infty \Big(r(\mathscr{R}^0 \cup \mathscr{R}^1) - r\left((\mathscr{R}^0 \cup \mathscr{R}^1)_{\le x}\right)\Big) dx - \int_0^\infty \Big(r(\mathscr{R}^0 \cup \mathscr{R}^1 \cup \mathscr{R}^2) - r\left((\mathscr{R}^0 \cup \mathscr{R}^1 \cup \mathscr{R}^2)_{\le x}\right)\Big) dx.
\end{aligned}$$

But using Equation (17), this gives precisely Equation (16). $\square$

We note that the proof of Lemma 21 easily generalizes to other matroids. This is a departure from the existing proofs in [19] and [4, Lemma 3.9], and Rizzi's more specific result [33, Lemma 2], although a strong *exchange property* of matroids is used in the proof of [4].

We are finally near the end of the analysis, where the Contraction Lemma comes into play. We can now bound the value $f_i(K^i)$ for all $0 \le i \le p-1$ in terms of the cost of T^*'s loss. In the remainder of the section, let the full components of T^* be $K^{*,1}, \dots, K^{*,q}$, let $\mathtt{l}^*$ denote $\mathtt{l}(T^*)$, let $\overline{\mathtt{mst}}^i$ denote $\overline{\mathtt{mst}}(\mathscr{S}^i)$ and let $\overline{\mathtt{mst}}^*$ denote $\overline{\mathtt{mst}}(T^*)$.

Lemma 23. *For all* $0 \le i \le p-1$, *if* $\overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^* > 0$, *then* $f_i(K^i) \le \mathtt{l}^*/(\overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^*)$.

Proof. By the choice of K^i in Algorithm 1, we have $f_i(K^i) \le \min_j f_i(K^{*,j})$. A standard fraction averaging argument implies that

$$\begin{aligned}
f_i(K^i) &\le \frac{\sum_{j=1}^q \mathtt{l}(K^{*,j})}{\sum_{j=1}^q \left(\overline{\mathtt{mst}}(\mathscr{S}^i) - \overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K^{*,j}\})\right)} \\
&\le \frac{\mathtt{l}^*}{\sum_{j=1}^q \left(\overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K^{*,1}, \dots, K^{*,j-1}\}) - \overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K^{*,1}, \dots, K^{*,j}\})\right)}
\end{aligned} \tag{20}$$

where the last inequality uses Fact 9 and Lemma 22. (Additional care is needed when T^* and $\mathscr{S}^p$ overlap in some full components, but the above inequalities still hold.) The denominator of the right-hand side of Equation (20) is a telescoping sum. Canceling like terms, and using Lemma 21 to replace $\overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K^{*,1}, \dots, K^{*,q}\})$ with $\overline{\mathtt{mst}}^*$, we are done. $\square$

We can now bound the cost of T^p.

Proof of Lemma 10. We first bound the loss $\mathtt{l}(T^p)$ of tree T^p. Using Fact 9,

$$\mathtt{l}(T^p) = \sum_{i=0}^{p-1} \mathtt{l}(K^i) = \sum_{i=0}^{p-1} f_i(K^i) \cdot (\overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^{i+1}) \tag{21}$$

where the last equality uses the definition of f_i from (10). Using Corollary 20 and Lemma 23, the right hand side of Equation (21) is bounded as follows:

$$\sum_{i=0}^{p-1} f_i(K^i)\cdot(\overline{\mathtt{mst}}^i-\overline{\mathtt{mst}}^{i+1}) \le \sum_{i=0}^{p-1} \frac{\mathtt{l}^*}{\max\{\mathtt{l}^*,\overline{\mathtt{mst}}^i-\overline{\mathtt{mst}}^*\}}\cdot(\overline{\mathtt{mst}}^i-\overline{\mathtt{mst}}^{i+1}). \tag{22}$$

The right hand side of Equation (22) can in turn be bounded from above by the following integral:

$$\sum_{i=0}^{p-1} \frac{\mathtt{l}^*\cdot(\overline{\mathtt{mst}}^i-\overline{\mathtt{mst}}^{i+1})}{\max\{\mathtt{l}^*,\overline{\mathtt{mst}}^i-\overline{\mathtt{mst}}^*\}} \le \int_{\overline{\mathtt{mst}}^p}^{\overline{\mathtt{mst}}^0} \frac{\mathtt{l}^*}{\max\{\mathtt{l}^*,x-\overline{\mathtt{mst}}^*\}}dx = \int_{\overline{\mathtt{mst}}^p-\overline{\mathtt{mst}}^*}^{\overline{\mathtt{mst}}^0-\overline{\mathtt{mst}}^*} \frac{\mathtt{l}^*}{\max\{\mathtt{l}^*,x\}}dx. \tag{23}$$

Notice that $\overline{\mathtt{mst}}^0=\mathtt{mst}(G[R],c)\ge\mathtt{opt}_r=\mathtt{l}^*+\overline{\mathtt{mst}}^*$. The termination condition in Algorithm 1 and Lemma 6 imply that $\overline{\mathtt{mst}}^p\le\mathtt{opt}_r$. Hence the result of evaluating the integral in the right-hand side of Equation (23) is

$$\mathtt{l}^*-(\overline{\mathtt{mst}}^p-\overline{\mathtt{mst}}^*)+\mathtt{l}^*\cdot\int_{\mathtt{l}^*}^{\overline{\mathtt{mst}}^0-\overline{\mathtt{mst}}^*}\frac{1}{x}dx=\mathtt{opt}_r-\overline{\mathtt{mst}}^p+\mathtt{l}^*\cdot\ln\left(\frac{\overline{\mathtt{mst}}^0-\overline{\mathtt{mst}}^*}{\mathtt{l}^*}\right) \tag{24}$$

where the equality uses Lemma 17. Applying Lemma 17 two more times, and combining Equations (21)–(24), we obtain

$$\begin{aligned} c(T^p)=\overline{\mathtt{mst}}^p+\mathtt{l}(T^p) \quad &\le \quad \mathtt{opt}_r+\mathtt{l}^*\cdot\ln\left(\frac{\overline{\mathtt{mst}}^0-\overline{\mathtt{mst}}^*}{\mathtt{l}^*}\right)\\ &= \quad \mathtt{opt}_r+\mathtt{l}^*\cdot\ln\left(1+\frac{\overline{\mathtt{mst}}^0-(\overline{\mathtt{mst}}^*+\mathtt{l}^*)}{\mathtt{l}^*}\right)\\ &= \quad \mathtt{opt}_r+\mathtt{l}^*\cdot\ln\left(1+\frac{\overline{\mathtt{mst}}^0-\mathtt{opt}_r}{\mathtt{l}^*}\right) \end{aligned}$$

as wanted. □

Remark. Gröpl et al. essentially prove Lemma 10 in [19, Lemma 4.3] but a minor error lies in their equation "(18)." Namely, they assume "$m_i-m^*>0$" which is $\overline{\mathtt{mst}}^i-\overline{\mathtt{mst}}^*>0$ in our notation.

References

[1] A. Agarwal and M. Charikar. On the advantage of network coding for improving network throughput. In *Proceedings, IEEE Information Theory Workshop*, 2004.

[2] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem in networks. *SIAM J. Comput.*, 24:440–456, 1995.

[3] Y. P. Aneja. An integer linear programming approach to the Steiner problem in graphs. *Networks*, 10:167–178, 1980.

[4] P. Berman and V. Ramaiyer. Improved approximations for the Steiner tree problem. *J. Algorithms*, 17(3):381–408, 1994.

[5] A. Borchers and D. Du. The *k*-Steiner ratio in graphs. *SIAM J. Comput.*, 26(3):857–869, 1997.

[6] M. Chlebík and J. Chlebíková. Approximation hardness of the Steiner tree problem on graphs. In *Proceedings, Scandinavian Workshop on Algorithm Theory*, pages 170–179, 2002.

[7] S. Chopra. On the spanning tree polyhedron. *Operations Research Letters*, 8:25–29, 1989.

[8] S. Chopra and M. R. Rao. The Steiner tree problem 1: Formulations, compositions, and extension of facets. *Mathematical Programming*, 64:209–229, 1994.

[9] S. Chopra and M. R. Rao. The Steiner tree problem 2: Properties and classes of facets. *Mathematical Programming*, 64:231–246, 1994.

[10] M. Didi Biha, H. Kerivin, and A. R. Mahjoub. Steiner trees and polyhedra. *Discrete Applied Mathematics*, 112(1-3):101–120, 2001.

[11] S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1:195–207, 1972.

[12] J. Edmonds. Optimum branchings. *J. Res. Nat. Bur. Standards*, B71:233–240, 1967.

[13] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP complete. *SIAM J. Appl. Math.*, 32:826–834, 1977.

[14] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM J. Appl. Math.*, 16(1):1–29, 1968.

[15] M. X. Goemans. The Steiner tree polytope and related polyhedra. *Math. Program.*, 63(2):157–182, 1994.

[16] M. X. Goemans and D. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Math. Programming*, 60:145–166, 1993.

[17] M. X. Goemans and Y. Myung. A catalog of Steiner tree formulations. *Networks*, 23:19–28, 1993.

[18] M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 4. PWS, Boston, 1997.

[19] C. Gröpl, S. Hougardy, T. Nierhoff, and H. J. Prömel. Approximation algorithms for the Steiner tree problem in graphs. In X. Cheng and D. Du, editors, *Steiner trees in industries*, pages 235–279. Kluwer Academic Publishers, Norvell, Massachusetts, 2001.

[20] S. Hougardy and H. J. Prömel. A 1.598 approximation algorithm for the Steiner problem in graphs. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms*, pages 448–453, 1999.

[21] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, NY, 1972.

[22] M. Karpinski and A. Zelikovsky. New approximation algorithms for the Steiner tree problems. *J. Combinatorial Optimization*, 1(1):47–65, 1997.

[23] J. Könemann, S. Leonardi, G. Schäfer, and S. van Zwam. From primal-dual to cost shares and back: A stronger LP relaxation for the Steiner forest problem. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 930–942. Springer, 2005.

[24] J. Könemann and D. Pritchard. Uncrossing partitions. Technical Report CORR 2007-11, University of Waterloo, 2007.

[25] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings, American Mathematical Society*, 7:48–50, 1956.

[26] T. Polzin and S. Vahdati Daneshmand. Primal-dual approaches to the Steiner problem. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(47), 2000.

[27] T. Polzin and S. Vahdati Daneshmand. A comparison of Steiner tree relaxations. *Discrete Applied Mathematics*, 112(1-3):241–261, 2001. Preliminary version appeared at COS 1998.

[28] T. Polzin and S. Vahdati Daneshmand. Improved algorithms for the Steiner problem in networks. *Discrete Applied Mathematics*, 112(1-3):263–300, 2001.

[29] T. Polzin and S. Vahdati Daneshmand. On Steiner trees and minimum spanning trees in hypergraphs. *Oper. Res. Lett.*, 31(1):12–20, 2003.

[30] H. J. Prömel and A. Steger. A new approximation algorithm for the Steiner tree problem with performance ratio 5/3. *J. Algorithms*, 36(1):89–101, 2000. Preliminary version appeared as "RNC-approximation algorithms for the Steiner problem" at STACS 1997.

[31] H. J. Prömel and A. Steger. *The Steiner Tree Problem — A Tour through Graphs, Algorithms, and Complexity*. Vieweg Verlag, Braunschweig-Wiesbaden, 2002.

[32] S. Rajagopalan and V. V. Vazirani. On the bidirected cut relaxation for the metric Steiner tree problem. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms*, pages 742–751, 1999.

[33] R. Rizzi. On Rajagopalan and Vazirani's $3/2$-approximation bound for the Iterated 1-Steiner heuristic. *Information Processing Letters*, 86(6):335–338, 2003.

[34] G. Robins and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM J. Discrete Math.*, 19(1):122–134, 2005. Preliminary version appeared as "Improved Steiner tree approximation in graphs" at SODA 2000.

[35] M. Skutella. Personal communication, 2006.

[36] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[37] D. Warme. A new exact algorithm for rectilinear Steiner trees. In P. Pardalos and D.-Z. Du, editors, *Network Design: Connectivity and Facilities Location: DIMACS Workshop April 28-30, 1997*, pages 357–395. American Mathematical Society, 1997. Preliminary version appeared at ISMP 1997.

[38] D. Warme. *Spanning Trees in Hypergraphs with Applications to Steiner Trees*. PhD thesis, University of Virginia, 1998.

[39] R. T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Math. Programming*, 28:271–287, 1984.

[40] A. Zelikovsky. Better approximation bounds for the network and Euclidean Steiner tree problems. Technical report, University of Virginia, Charlottesville, VA, USA, 1996.

[41] A. Z. Zelikovsky. An $11/6$-approximation algorithm for the network Steiner problem. *Algorithmica*, 9:463–470, 1993.